Toward a Timed Theory of Channel Coding*

Eugene Asarin[1], Nicolas Basset[2], Marie-Pierre Béal[2],
Aldric Degorre[1], and Dominique Perrin[2]

[1] LIAFA, University Paris Diderot and CNRS, France
[2] LIGM, University Paris-Est Marne-la-Vallée and CNRS, France

Abstract. The classical theory of constrained-channel coding deals with the following questions: given two languages representing a source and a channel, is it possible to encode source messages to channel messages, and how to realize encoding and decoding by simple algorithms, most often transducers. The answers to this kind of questions are based on the notion of entropy.
In the current paper, the questions and the results of the classical theory are lifted to timed languages. Using the notion of entropy of timed languages introduced by Asarin and Degorre, the question of timed coding is stated and solved in several settings.

1 Introduction

This paper is the first attempt to lift the classical theory of constrained-channel coding to timed languages.

Let a language S represent all the possible messages that can be generated by a *source*, and C all the messages that can transit over a *channel*. Typical problems addressed by coding theory are:

- Is it possible to transmit any source generated message via the channel?
- What would be the transmission speed?
- How to encode the message before and to decode it after transmission?

The answers given by the theory of channel coding are as follows: to each language L is associated a non-negative real number $h(L)$, called its entropy, which characterizes the quantity of information in bits per symbol. In order to transmit information in real-time (resp. with speed α) the entropy of the source should not exceed the one of the channel: $h(S) \leq h(C)$ (resp. $\alpha h(S) \leq h(C)$). For regular (or more precisely sofic) languages, whenever the information inequalities above are strict, the theory of channel coding provides a transmission protocol with a simple encoding and decoding (realized by a finite-state transducer). For the practically important case when $S = \Sigma^*$ and $h(S) < h(C)$, the decoding can be made even simpler (sliding-window). A typical example is EFMPlus code [12] allowing writing any binary file (i.e. the source $\{0,1\}^*$, with entropy 1) onto a

* The support of Agence Nationale de la Recherche under the project EQINOCS (ANR-11-BS02-004) is gratefully acknowledged.

DVD (the channel $C = (2, 10) - \text{RLL}$ admits all the words without factors 11, 101 and 0^{11}, its entropy is 0.5418, see [9]) with almost optimal rate $\alpha = 1/2$.

Classical theory of channel coding deals with discrete messages. It is, however, important to consider data words, i.e. discrete words augmented with data, e.g. real numbers. In this paper, we develop the theory of channel coding for the most studied class of data languages: timed languages. Several models of information transmission are possible for the latter:

- the source is a timed language; the channel is a discrete language. In this case, lossless encoding is impossible, and we will consider encoding with some precision ε;
- the source and the channel are timed languages, we are interested in exact (lossless) encoding;
- the source and the channel are timed languages, some scaling of time data is allowed.

Our solution will be based on the notion of entropy of timed languages [3]. For several models of transmission of timed data we will write *information inequalities* relating entropies of sources and channels with parameters of encodings (rate, precision, scaling, see below). Such an inequality is a necessary condition for existence of an encoding. On the other hand, whenever the information inequality holds (in its strict form) and the languages are regular (sofic), we give an explicit construction for simple timed encoding-decoding functions.

Related work. Constrained-channel coding theory for finite alphabets is a well-established domain; we refer the reader to monographs [13, 9, 8], handbook chapters [14, 7] and references therein. We started exploring information contents of timed languages in [2–4] where the notion of entropy was introduced and related to information measures such as Kolmogorov complexity and ε-entropy. Technically, we strongly build on discretization of timed languages, especially [3, 6]. In a less formal way, a vision of timed languages as a special kind of languages with data [11, 10] was another source of inspiration.

Paper structure. In Sect. 2 we recall basic notions of the discrete theory of constrained-channel coding. In Sect. 3 we briefly recall some results and constructions on volume, entropy and discretization of timed languages. In Sect. 4 we state our main results on timed theory of constrained-channel coding. In Sect. 5 we discuss the rationale, perspectives and applications of this work.

2 Theory of channel coding for finite alphabet languages

In this section we give an elementary exposition[3] of some basic notions and results from the theory of constrained-channel coding, see [14, 13, 8, 7] for more details.

[3] We avoid here the terminology of symbolic dynamics, standard in the area of coding.

2.1 Terminology

Let Σ be a finite alphabet. A *factor* of a word $w \in \Sigma^*$ is a contiguous subsequence of its letters. A language L is *factorial* whenever for each word $w \in L$ every factor of it is in the language. A language L is *extensible* whenever for each word $w \in L$ there exists a letter $a \in \Sigma$ such that $wa \in L$.

These two conditions are usual in the context of coding and can be justified in practice as follows. If we can encode (decode) some long word w (e.g. a movie file), then we want also to encode (decode) its contiguous fragments (e.g. a short scene in the middle of the movie). On the other hand, some extension of w should correspond to a longer movie.

A language L, which is regular, factorial and extensible, is called *sofic*. Sofic languages can be recognized by finite automata whose states are all initial and final (this ensures factoriality) and all have outgoing transitions (this ensures extensibility).

Given a language L, we denote by L_n its sublanguage of words of length n. The *entropy* $h(L)$ of a language over a finite alphabet is the asymptotic growth rate of the cardinality of L_n, formally defined by (all the logarithms are base 2):

$$h(L) = \lim_{n\to\infty} \frac{1}{n} \log |L_n|.$$

The limit (finite or $-\infty$) always exists if L is factorial. For a sofic language L recognized by a given automaton, its entropy $h(L)$ can be effectively computed using linear algebra. In particular if $L = \Sigma^*$ for a k-letter alphabet Σ then $h(L) = \log k$. Finally, the intuitive meaning of the entropy is the amount of information (in bits per symbol) in typical words of the language.

Most of our coding functions have a special property defined below.

Definition 1 (almost injective). *A (partial) function $\phi : \Sigma^* \to \Gamma^*$ is called* almost injective *with delay $d \in \mathbb{N}$, if for any n and $w, w' \in \Sigma^n$, and $u, u' \in \Sigma^d$ it holds that*

$$\phi(wu) = \phi(w'u') \Rightarrow w = w'.$$

Intuitively, if such a function is used to encode messages, then knowing the code of some message wu one can decode w, i.e. the whole message except its last d symbols. Thus the decoding is possible with delay d. This can be formalized as follows:

Definition 2 (almost inverse). *For an almost injective function $\phi : \Sigma^* \to \Gamma^*$ with delay d its d-*almost inverse *family of functions $\psi_n : \Gamma^* \to \Sigma^n$ is characterized by the following property: for any $w \in \Sigma^n$ and $v \in \Gamma^*$,*

$$w = \psi_n(v) \Leftrightarrow \exists u \in \Sigma^d : \ \phi(wu) = v.$$

Lemma 1. *If the domain of an almost injective function ϕ is extensible and ψ_n is its almost inverse, then ψ_n is a surjection to this domain (constrained to words of length n).*

2.2 Coding: the basic case

Let A and A' be two alphabets (source and channel alphabets), $S \subset A^*$ and $C \subset A'^*$ factorial extensible languages and $d \in \mathbb{N}$. The aim is to encode any source message $w \in S$ to a channel message $\phi(w) \in C$. The latter message can be transmitted over the channel.

Definition 3. *An (S, C)-encoding with delay d is a function $\phi : S \to C$ (total on S but not necessarily onto C) such that*

- *it is length preserving: $\forall w \in S,\ |\phi(w)| = |w|$,*
- *it is almost injective with delay d.*

The first condition means that the information is transmitted in real-time (with the transmission rate 1). The second one permits decoding.

A natural question is to find necessary and sufficient conditions on S and C for an (S, C)-encoding (with some delay) to exist. This question can be addressed by comparing the entropy of the languages S and C. Roughly, the channel language should contain at least as much information per symbol as the source language. Formally, we define the *information inequality*:

$$h(S) \leq h(C). \tag{II1}$$

Proposition 1. *Let S and C be factorial and extensible languages. If an (S, C)-encoding exists then* (II1) *necessarily holds.*

Proof. Let $\phi : S \to C$ be an (S, C)-encoding with delay d. By Lemma 1 its almost inverse ψ maps C onto S. More precisely, for every n we have $\psi_n(C_{n+d}) = S_n$. Hence, the cardinalities should satisfy: $|S_n| \leq |C_{n+d}|$. Finally we have

$$\lim_{n\to\infty} \frac{1}{n} \log |S_n| \leq \lim_{n\to\infty} \frac{1}{n} \log |C_{n+d}|$$

and the expected inequality $h(S) \leq h(C)$. □

Thus, (II1) is necessary for existence of the coding. For sofic languages (if the inequality is strict) it is also sufficient. Moreover, the encoding can be realized by a sort of finite-state machine. We present this fundamental result in the following form which is essentially the finite-state coding theorem from [14], Theorem 10.3.7 in [13].

Theorem 1. *Let S and C be sofic languages. If the strict version of* (II1) *holds, then there exists an (S, C)-encoding realized by a finite-state transducer which is right-resolving on input and right-closing on output*[4]*.*

The reader is now motivated to get through a couple of definitions.

Definition 4 (transducer). *A transducer is a tuple $\tau = (Q, A, A', \Delta, \mathcal{I}, \mathcal{O})$ with a finite set Q of control states; finite input and output alphabets A and A'; a set of transitions Δ (each transition δ has a starting state $\mathtt{ori}(\delta)$ and an ending state $\mathtt{ter}(\delta)$); input and output labeling functions $\mathcal{I} : \Delta \to A$ and $\mathcal{O} : \Delta \to A'$.*

[4] Such a transducer is also called a finite-state (S, C)-encoder.

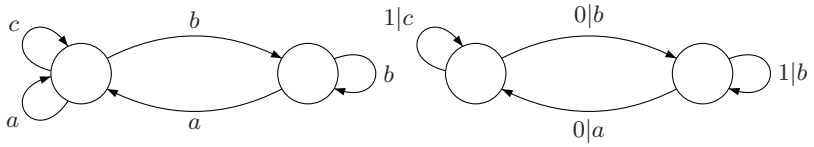

Fig. 1. A sofic automaton of a channel C and a $(\{0,1\}^*, C)$-encoder

The transducer is said to be *right-resolving* on input whenever for each state $q \in Q$ every two different edges starting from q have different input labels. For such a transducer, the input automaton with a fixed initial state i, i.e. $\mathcal{A}_i = (Q, A, \Delta, \mathcal{I}, i, Q)$ is deterministic, we denote by S_i the language of this automaton.

The transducer is *right-closing* on output with delay d whenever every two paths π and π' of length $d+1$ with the same output label and the same starting state q always have the same initial edge $\pi_1 = \pi'_1$.

A transducer τ satisfying both properties performs the encoding process in a natural way: an input word w of S is read from a state i along a path π_w, and this path determines the output word $\mathcal{O}(\pi_w)$. The function $\phi_i : S_i \to A'^*$ defined as $w \mapsto \mathcal{O}(\pi_w)$ is length preserving and almost injective with delay d.

Example 1. Consider the source language $S = \{0,1\}^*$ and the channel language C recognized by the sofic automaton on the left of Fig. 1. The language C is composed by all the words on $\{a, b, c\}$ that do not contain any block bc. The entropy of the source is $h(S) = 1$, and the one of the channel is $h(C) = 2\log\left[(1+\sqrt{5})/2\right] \approx 1.3885$. The information inequality $h(S) < h(C)$ holds and we can encode S in C using the transducer on the right of Fig. 1.

2.3 Other coding settings

Similarly to the previous section, other coding settings can be considered. For example, we can transmit information over a channel with some rate $\alpha = \frac{p}{q}$, when q letters of the channel message correspond to p letters of the source message (the previous section corresponds thus to the case $\alpha = 1$).

Definition 5. *An (S, C)-encoding with rate $\alpha \in \mathbb{Q}^+$ and delay d is a function $\phi : S \to C$ (total on S and not necessarily onto C) such that*

- *it is of rate α, i.e. $\forall w \in S,\ \lceil \alpha |\phi(w)| \rceil = |w|$;*
- *it is almost injective (with delay d).*

In this setting, the information inequality takes the form:

$$\alpha h(S) \leq h(C), \tag{II2}$$

and it is a necessary and almost sufficient condition for the code to exist:

Proposition 2. *Let S and C be factorial and extensible languages. If an (S, C)-encoding with rate α exists, then* (II2) *necessarily holds.*

Proposition 3. *Let S and C be sofic languages. If a strict inequality* (II2) *holds, then an (S, C)-encoding of rate α exists. Moreover, it can be realized by a finite-state transducer of rate α.*

We skip here a natural definition of such a transducer.

3 Preliminaries on timed languages

3.1 Timed alphabets, words, languages, and automata

We call *k-M-alphabet* a set $A = [0, M] \times \Sigma$ where Σ is a k-letter alphabet and M a positive integer bound, so that every letter in A corresponds to a real-valued delay (seen as data) in $[0, M]$ and a discrete event in Σ. Timed words and languages are respectively words and languages over a k-M-alphabet. We define factor-closed and extensible timed language as in the untimed case.

Timed automata as described below can be used to define timed languages, which are called *regular*. First we note $G_{c,M}$ the set of all c-dimensional M-bounded rectangular integer guards, i.e. Cartesian products of c real intervals I_i, $i \in \{1..c\}$, having integer bounds in $\{0..M\}$. A *timed automaton* is thus a tuple $\mathcal{A} = (Q, c, M, \Sigma, \Delta, I, F)$ where Q is a finite set of locations; $c \in \mathbb{N}$ is the number of clocks; $M \in \mathbb{N}$ is an upper bound on clock values; Σ is a finite alphabet of events; $\Delta \subseteq Q \times Q \times \Sigma \times G_{c,M} \times 2^{\{1..c\}}$ is a set of transitions; $I : Q \to G_{c,M}$ maps each location to an initial constraint; $F : Q \to G_{c,M}$ maps each location to a final constraint. A transition $\delta = \langle \mathtt{ori}(\delta), \mathtt{ter}(\delta), \mathcal{L}(\delta), \mathfrak{g}(\delta), \mathfrak{r}(\delta) \rangle \in \Delta$ is such that $\mathtt{ori}(\delta)$ is its origin location, $\mathtt{ter}(\delta)$ is its ending location, $\mathcal{L}(\delta)$ is its label, $\mathfrak{g}(\delta)$ is the guard that clock values must satisfy for firing δ and $\mathfrak{r}(\delta)$ is the set of clocks reset by firing it.

A timed word $(t_1, a_1) \dots (t_k, a_k)$ is in the language of a timed automaton if there exists a run $(q_i, \mathbf{x}_i)_{i \in \{0..k\}}$ with $q_i \in Q$ and $\mathbf{x}_i \in [0, M]^c$, such that $\mathbf{x}_0 \in I(q_0)$, $\mathbf{x}_k \in F(q_k)$ and such that for all i, there exists $\delta_i \in \Delta$ satisfying $\mathtt{ori}(\delta_i) = q_{i-1}$, $\mathtt{ter}(\delta_i) = q_i$, $\mathcal{L}(\delta_i) = a_i$, $\mathbf{x}_{i-1} + t_i \in \mathfrak{g}(\delta_i)$ and $\mathbf{x}_i = \mathfrak{r}(\delta_i)(\mathbf{x}_{i-1} + t_i)$ (i.e. equal to $\mathbf{x}_{i-1} + t_i$ where coordinates in $\mathfrak{r}(\delta_i)$ are substituted by zeros).

A timed automaton is said to be right-resolving if outgoing transitions from the same location with the same label have pairwise incompatible guards. If we add the condition of having only one initial state we obtain the classical definition of determinism of [1]. Languages recognized by right-resolving timed automata are also said right-resolving. Right-resolving factor-closed and extensible timed regular languages are called *sofic*.

An example of timed automaton is given in Fig. 2.

3.2 Volume and entropy

From now on A denotes a k-M-alphabet $[0, M] \times \Sigma$. Let $L \subseteq A^n$ be a timed language and $n \in \mathbb{N}$, we denote by L_n the set of timed words of length n

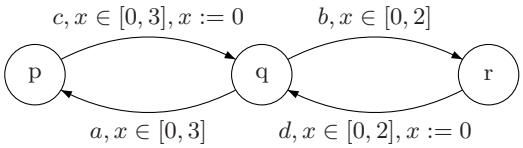

Fig. 2. A timed automaton (all the states are initial and final).

in L. For each $w = w_1 \dots w_n \in \Sigma^n$ we denote by L_w the (possibly empty) set of points $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{R}^n$ such that $(t_1, w_1) \dots (t_n, w_n) \in L_n$. Thus $L_n = \biguplus_{w \in \Sigma^n} L_w \times \{w\}$ (by a slight abuse of notation).

The language L is said to be measurable whenever for all $w \in \Sigma^*$, L_w is Lebesgue measurable, i.e. its (hyper-)volume $\mathtt{Vol}(L_w)$ is well defined. The volume is just the n-dimensional generalization of interval length in $\mathbb{R}$, area in $\mathbb{R}^2$, volume in $\mathbb{R}^3$,... Examples of timed languages and their volume are given later, we also refer to the papers [3, 6] for more detailed examples. For a timed regular language L, languages L_w ($w \in \Sigma^*$) are just finite unions of convex polytopes (see for instance Fig. 4) and hence measurable [3]. In the sequel, all timed languages considered are measurable. The *volume* of L_n and the *volumetric entropy* of L are defined respectively as

$$\mathtt{Vol}(L_n) = \sum_{w \in \Sigma^n} \mathtt{Vol}(L_w); \quad \mathcal{H}(L) = \lim_{n \to \infty} \frac{1}{n} \log \mathtt{Vol}(L_n),$$

and it can be shown that the limit exists for any factorial language.

For our running example on Fig. 2, $L_{ac} = \{(t_1, t_2) \mid t_1 + t_2 \leq 3\}$ and $L_{bd} = \{(t_1, t_2) \mid t_1 + t_2 \leq 2\}$; and their volumes are respectively 4.5 and 2; and thus $L_2(q)$, the sublanguage of L_2 of words accepted by runs starting from q, has volume 6.5. We have $L_{2n}(q) = (L_2(q))^n$ whose volume is 6.5^n, the entropy of the whole language $\mathcal{H}(L)$ is thus at least $0.5 \log 6.5$ (in fact it is exactly $0.5 \log 6.5$).

3.3 Discretization of languages and entropy

Here we adapt some definitions and results from [3, 6]. For a k-M-alphabet $A = [0, M] \times \Sigma$, we denote by A_ε its ε discretization: $A_\varepsilon = \{0, \varepsilon, 2\varepsilon, \dots, M\} \times \Sigma$. We remark that A_ε is a finite alphabet of size $k(\frac{M}{\varepsilon} + 1)$. An ε-discrete word is a timed word whose timed delays are multiples of ε. Given a timed language $L \subseteq A^*$, its ε-discretization L_ε is the discrete language on A_ε composed by ε-discrete words in L: $L_\varepsilon = L \cap {A_\varepsilon}^*$.

Given an ε-discrete word $w = (t_1, a_1) \dots (t_n, a_n)$, its ε-North-East-neighborhood is the timed set $\mathcal{B}_\varepsilon^{NE}(w) = \{(u_1, a_1) \dots (u_n, a_n) \mid u_i \in [t_i, t_i + \varepsilon],\ i = 1..n\}$. We extend this definition to associate timed languages with languages of ε-discrete words $\mathcal{B}_\varepsilon^{NE}(L_\varepsilon) = \cup_{w \in L_\varepsilon} \mathcal{B}_\varepsilon^{NE}(w)$.

We will use discretization in a three-step reduction scheme:

1. discretize the timed languages S, C with a sampling rate ε to obtain $S_\varepsilon, C_\varepsilon$;
2. use classical coding theorem 1 with $S_\varepsilon, C_\varepsilon$;
3. go back to timed languages by taking ε-NE-neighborhood of S_ε and C_ε.

The following lemma is the main tool for this reduction scheme, it is a variant of results of [3, 6]:

Lemma 2. *Let S be a timed sofic language. If $\mathcal{H}(S) > -\infty$ then for all positive small enough ε, one can compute ε-discrete sofic languages S_ε^- and S_ε^+ that verify $\mathcal{B}_\varepsilon^{NE}(S_\varepsilon^-) \subseteq S \subseteq \mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+)$, and $\mathcal{H}(S) + \log \frac{1}{\varepsilon} = h(S_\varepsilon^-) + o(1) = h(S_\varepsilon^+) + o(1)$.*

4 Timed coding

Similarly to classical results presented in Sect. 2, we will consider several settings for transmission of timed words over a channel. For every setting we will formulate an information inequality, and show that it is necessary and, with some additional hypotheses, sufficient for a coding to exist.

4.1 Timed source, discrete channel, approximate transmission

In practice, timed and data words are often transmitted via discrete (finite alphabet) channels. For example, a timed log of events in an operating system (a timed message) can be stored as a text file (ASCII message). The delays in the timed word cannot be stored with infinite precision, thus the coding is necessarily approximate. More precisely, the set of timed source messages w of a length n is infinite, while the set of discrete channel messages of the same length is finite. For this reason, the coding cannot be injective, and necessarily maps many timed words to a same discrete word. It is natural to require that all the timed words with the same code are ε-close to each other. This justifies Def. 6 below. We give first some notation. For two timed words of same length $w = (t_1, a_1) \dots (t_n, a_n)$ and $w = (t'_1, a'_1) \dots (t'_n, a'_n)$, the distance $\mathtt{dist}(w, w')$ between w and w' is equal to $+\infty$ if $a_1 \dots a_n \neq a'_1 \dots a'_n$, otherwise it is $\max_{1 \leq i \leq n} |t_i - t'_i|$. Let A be a k-M alphabet, Σ' be a finite alphabet, S be a factorial extensible measurable timed language on A, C be a factorial extensible language on Σ', and α, ε be positive reals and d be a non negative integer.

Definition 6. *Similarly to Def. 1 we say that a partial function $\phi : A^* \to \Sigma'^*$ is* almost approximately injective *with precision ε and delay d if*

$$\forall n \in \mathbb{N}, w, w' \in A^n \ \forall u, u' \in A^d : \ \phi(wu) = \phi(w'u') \Rightarrow \mathtt{dist}(w, w') < \varepsilon.$$

Its almost inverse *is a multi-valued function family $\psi_n : \Sigma'^* \to A^n$ characterized by the following property: for any $w \in A^n$ it holds that $w \in \psi_n(v)$ if and only if some $u \in A^d$ yields $\phi(wu) = v$.*

Lemma 3. *Given an almost approximately injective function ϕ with precision ε and delay d, let ψ_n be its almost inverse family. Then for every v the diameter of $\psi_n(v)$ is at most ε. If the domain of ϕ is extensible, then the image of ψ_n coincides with this domain (constrained to length n).*

Definition 7. *An (S, C)-encoding of a rational rate α, precision ε and delay d is a function $\phi : S \to C$ (total on S) such that*

- *it is of rate α: i.e. $\forall w \in S,\ \lceil \alpha |\phi(w)| \rceil = |w|$;*
- *it is almost approximately injective with precision ε and delay d.*

The information inequality for this setting has the form:

$$\alpha(\mathcal{H}(S) + \log(1/\varepsilon)) \leq h(C), \tag{II3}$$

which corresponds to information contents of S equal to $\mathcal{H}(S) + \log(1/\varepsilon)$, see the formula for Kolmogorov complexity of timed words in [3].

Proposition 4. *For a factorial extensible measurable timed language S and a factorial extensible discrete language C the following holds. If an (S, C)-encoding of rate α, precision ε, and some delay d exists then necessarily* (II3) *must be satisfied.*

Proof. Let ϕ be an (S, C)-encoding of rate α, precision ε and delay d, and ψ its almost inverse. By Lemma 3, for every n it holds that $S_n = \psi_n(C_{\lfloor (n+d)/\alpha \rfloor})$. This leads to an inequality on volumes

$$\mathtt{Vol}(S_n) \leq \sum_{v \in C_{\lfloor (n+d)/\alpha \rfloor}} \mathtt{Vol}(\psi_n(v)).$$

Any $\psi(v)$ has a diameter $\leq \varepsilon$ and thus is included in a cube of side ε and volume ε^n. We have:

$$\mathtt{Vol}(S_n) \leq \varepsilon^n |C_{\lfloor (n+d)/\alpha \rfloor}|.$$

Thus

$$\frac{\alpha}{n} \log \mathtt{Vol}(S_n) \leq \frac{\alpha}{n} \log \varepsilon^n |C_{\lfloor (n+d)/\alpha \rfloor}| = \alpha \log \varepsilon + \frac{\lfloor (n+d)/\alpha \rfloor}{n/\alpha} \frac{|C_{\lfloor (n+d)/\alpha \rfloor}|}{\lfloor (n+d)/\alpha \rfloor}.$$

Taking the limit as n tends to infinity we obtain $\alpha \mathcal{H}(S) \leq \alpha \log \varepsilon + h(C)$ and then (II3) holds. □

We strengthen a little bit (II3) to have a (partial) converse result for sofic timed languages.

Proposition 5. *For a sofic timed language S with $\mathcal{H}(S) > -\infty$, there exists a function R_S such that $\lim_{x \to 0} R_S(x) = 0$ and the following holds. Whenever the entropy of a sofic discrete language C verifies the inequality $\alpha(\mathcal{H}(S) + \log(1/\varepsilon) + R_S(\varepsilon)) < h(C)$, then there exists an (S, C)-encoding of rate α, precision ε and some delay d. Moreover it can be realized by a "real-time transducer" sketched below in the proof.*

Proof (Sketch). Let S be a sofic timed language. For $\varepsilon > 0$, let S_ε^+ be its ε-discretized over-approximation given by Lemma 2. We define

$$R_S(\varepsilon) = h(S_\varepsilon^+) - \mathcal{H}(S) - \log(1/\varepsilon),$$

it satisfies the required condition: $R_S(\varepsilon) = o(1)$ (see Lemma 2). Let C be a sofic discrete language such that

$$\alpha(\mathcal{H}(S) + \log(1/\varepsilon) + R_S(\varepsilon)) < h(C),$$

we prove that an (S, C)-encoding of rate α, precision ε and some delay d exists. Lemma 2 gives us

$$S \subseteq \mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+) \text{ and } \alpha h(S_\varepsilon^+) = \alpha(\mathcal{H}(S) + \log(1/\varepsilon) + R_S(\varepsilon)) < h(C).$$

Thus by Prop. 3 an (S_ε^+, C)-encoding of rate α and some delay d exists and can be realized by a finite-state transducer τ_ε of rate α and delay d. If we replace for each transition its input label $(a, k\varepsilon)$ by the label a and the guard $t \in [k\varepsilon, (k+1)\varepsilon]$, we obtain a real-time transducer τ with input $\mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+)$ whose output is in C. The injectivity of τ_ε ensures that τ realizes an approximately injective function with precision ε. □

4.2 Timed source, timed channel, exact transmission

Another natural setting is when a timed message is transmitted via a timed channel. In this case, the coding can be exact (injective). For the moment we consider length-preserving transmission (see Sect. 4.4 for faster and slower transmission).

Let A, A' be a k-M and a k'-M' alphabet, S and C factorial extensible measurable timed languages on these alphabets, and $d \in \mathbb{N}$.

Let ℓ and σ be positive rationals. A function $f : A'^n \to A^*$ is said to be ℓ-Lipshitz whenever for all x, y in its domain, $\mathtt{dist}(f(x), f(x')) \leq \ell\, \mathtt{dist}(x, x')$. We call a function σ-piecewise ℓ-Lipshitz if its restriction to each cube of the standard σ-grid on A'^n is ℓ-Lipshitz.

We can now state the definition of an (S, C)-encoding:

Definition 8. *An (S, C)-encoding with delay d (and step σ) is a function $\phi : S \to C$ such that*

- *it is length preserving: $|\phi(w)| = |w|$,*
- *it is almost injective (with delay d),*
- *no time scaling: the almost inverse ψ_n are σ-piecewise 1-Lipshitz.*

The last condition rules out a possible cheating when all the time delays are divided by 1000 before transmission over the channel. We will come back to this issue in Sect. 4.3.

The information inequality in this setting takes a very simple form:

$$\mathcal{H}(S) \leq \mathcal{H}(C). \tag{II4}$$

The necessary condition for existence of a coding has a standard form (for technical reasons we require the channel to be sofic):

Proposition 6. *If for a factorial extensible measurable timed language S and a sofic timed language C an (S, C)-encoding of delay d exists, then* (II4) *holds.*

Proof. We consider first the most interesting case when $\mathcal{H}(C) > -\infty$. We will prove that for any $\zeta > 0$ the inequality $\mathcal{H}(S) \leq \mathcal{H}(C) + \zeta$ holds. Suppose ϕ is an (S, C)-encoding of delay d (and step σ), and ψ its almost inverse. By Lemma 1, for any natural n we have $S_n \subset \psi_n(C_{n+d})$.

Since C is sofic, Lemma 2 applies, and for a fixed ϵ, each C_n can be covered by C_n^+, a union of $K_{n,\varepsilon}$ cubes of size ε with $\mathcal{H}(C) + \log \frac{1}{\varepsilon} = \lim_{n\to\infty} \frac{\log K_{n,\varepsilon}}{n} + o(1)$. We choose ε dividing σ and small enough such that

$$\mathcal{H}(C) + \log \frac{1}{\varepsilon} > \lim_{n\to\infty} \frac{\log K_{n,\varepsilon}}{n} - \zeta. \tag{1}$$

Thus we have $S_n \subset \psi_n(C^+_{n+d})$, and, passing to volumes we get

$$\mathtt{Vol}(S_n) \leq \mathtt{Vol}(\psi_n(C^+_{n+d})) \leq K_{n+d,\varepsilon}\varepsilon^n, \tag{2}$$

indeed, since ψ_n is 1-Lipshitz on each ε-cube, ψ_n-image of each such cube has a diameter $\leq \varepsilon$ and thus a volume $\leq \varepsilon^n$. Passing to logarithms, dividing by n and taking the limit as $n \to \infty$ in (2) we get

$$\mathcal{H}(S) \leq \lim_{n\to\infty} \frac{\log K_{n+d,\varepsilon}}{n+d} + \log \varepsilon,$$

and applying inequality (1) we obtain

$$\mathcal{H}(S) \leq \mathcal{H}(C) + \log \frac{1}{\varepsilon} + \zeta + \log \varepsilon = \mathcal{H}(C) + \zeta,$$

which concludes the proof for the case when $\mathcal{H}(C) > -\infty$. The remaining case $\mathcal{H}(C) = -\infty$ is a simple corollary of the previous one. □

As usual, when both timed languages S and C are sofic and the information inequality (II4) strict, the converse holds.

Proposition 7. *If for sofic timed languages S and C it holds that $H(S) < H(C)$, then there exists an (S, C)-encoding (with some delay d). Moreover it can be realized by a "real-time transducer" described below in the proof.*

Proof (Sketch). Let S and C be sofic timed languages whose entropies verify $-\infty < \mathcal{H}(S) < \mathcal{H}(C)$. We prove that an (S, C)-encoding with some delay d exists. Let C_ε^- and S_ε^+ be as in Lemma 2, i.e. such that

$$S \subseteq \mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+);\ \mathcal{B}_\varepsilon^{NE}(C_\varepsilon^-) \subseteq C;$$
$$\mathcal{H}(S) + \log \frac{1}{\varepsilon} = h(S_\varepsilon^+) + o(1);\ \mathcal{H}(C) + \log \frac{1}{\varepsilon} = h(C_\varepsilon^-) + o(1).$$

The discretization step ε can be chosen small enough such that $h(S_\varepsilon^+) < h(C_\varepsilon^-)$. Thus by Theorem 1 a finite-state $(S_\varepsilon^+, C_\varepsilon^-)$-encoder τ_ε exists.

We replace each transition δ_ϵ of τ_ϵ with input label $(a, k\varepsilon)$ and output label $(b, l\varepsilon)$ by a transition δ with input label a, guards $t \in [k\varepsilon, (k+1)\varepsilon]$, output

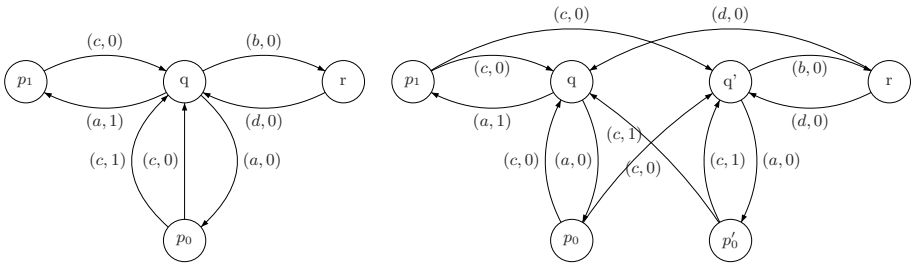

Fig. 3. Left: $\mathcal{A}_\varepsilon^-$: an automaton recognizing C_ε^- with $\varepsilon = 1$. Right: its split form.

label b and increment/decrement $c(\delta) = (l-k)\varepsilon$. We obtain what we call a *real-time transducer* τ. Its input language is $\mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+)$ and its output language is included in $\mathcal{B}_\varepsilon^{NE}(C_\varepsilon^-) \subseteq C$. The encoding is performed as follows: a timed word $(t_1, a_1)\dots(t_n, a_n)$ is in the input language if there is a path $\delta_1 \dots \delta_n$ such that $\mathcal{I}(\delta_i) = a_i$ and t_i satisfies the guard of δ_i: $t_i \in [k\varepsilon, (k+1)\varepsilon]$; the output timed word is in this case $(t'_1, b_1)\dots(t'_n, b_n)$ with $t'_i = t_i + c(\delta_i)$, $b_i = \mathcal{O}(\delta_i)$.

The collection of cubes $\mathcal{B}_\varepsilon^{NE}(w)$, $w \in S_\varepsilon^+$ (resp. $w \in C_\varepsilon^-$) forms a partition of timed languages $\mathcal{B}_\varepsilon^{NE}(S_\varepsilon^+)$ (resp. $\mathcal{B}_\varepsilon^{NE}(C_\varepsilon^-)$), they are cubes of the standard σ-grid with the step $\sigma = \epsilon$. Transducer τ only translates cubes. Translations are 1-Lipshitz and thus the last condition of an (S, C)-encoding holds.

The remaining degenerate case when $-\infty = \mathcal{H}(S) < \mathcal{H}(C)$ is an easy corollary of the non-degenerate one. □

The following example illustrates the construction of the transducer. Let the source timed language be $S = ([0,1] \times \{e, f\})^*$ and the channel timed language C be recognized by the automaton on Fig. 2. We have seen that the entropy $\mathcal{H}(C)$ is at least $0.5 \log 6.5$ and thus $\mathcal{H}(C) > \mathcal{H}(S) = \log 2$. By Prop. 7 an (S, C)-encoding exists. To realize this encoding we take $\varepsilon = 1$. There are four cubes included in the language $C_2(q)$: $([0,1] \times [0,1] \times \{ac\}, [0,1] \times [0,2] \times \{ac\}, [1,0] \times [0,1] \times \{ac\}, [0,1] \times [0,1] \times \{bd\})$ while the cubes to encode (language S_2) are $[0,1] \times \{ee\}, [0,1] \times \{ef\}, [0,1] \times \{fe\}, [0,1] \times \{ff\}$. The transducer will repeatedly map four "input cubes" to four "output cubes". We build an automaton for discrete words C_ε^- (as in Lemma 2, such words correspond to "output cubes") in Fig. 3, left. Then, as usual in coding, we first split the state p_0 and then the state q (each in two copies) to obtain an automaton with constant outdegree 2 (Fig. 3, right). This automaton accepts the same language C_ε^-, and can be transformed

$\mathtt{ori}(\delta)$	$\mathtt{ter}(\delta)$	$\mathcal{I}(\delta)$	$\mathfrak{g}(\delta)$	$\mathcal{O}(\delta)$	$c(\delta)$
q	p_0	e	$[0,1]$	a	0
q	p_1	f	$[0,1]$	a	1
q'	p'_0	e	$[0,1]$	a	0
q'	r	f	$[0,1]$	b	1
p_0	q	e	$[0,1]$	c	0
p_0	q'	f	$[0,1]$	c	0
p'_0	q	e	$[0,1]$	c	1
p'_0	q'	f	$[0,1]$	c	1
p_1	q	e	$[0,1]$	c	0
p_1	q'	f	$[0,1]$	c	0
r	q	e	$[0,1]$	d	0
r	q'	f	$[0,1]$	d	0

Table 1. The coding transducer

to the desired transducer just by adding input letters and increment/decrement to its transition. The transitions of the transducer are given in Table 1.

4.3 A variant: scaling allowed

In some situations, timed data can be scaled for coding, which leads to a new term in the information inequality. Let $\lambda > 0$ be a rational bound on scaling factor. We modify Def. 8 by replacing 1-Lipshitz by $1/\lambda$-Lipshitz.

Definition 9. *An (S, C)-encoding with delay d, scaling λ and step σ is a function $\phi : S \to C$ such that*

- *it is length preserving: $|\phi(w)| = |w|$,*
- *it is almost injective (with delay d),*
- *it has scaling at most λ: the almost inverse ψ_n are σ-piecewise $1/\lambda$-Lipshitz.*

The information inequality for this case becomes:

$$\mathcal{H}(S) \leq \mathcal{H}(C) + \log \lambda. \tag{II5}$$

The problem of coding with scaling can be easily reduced to the one considered in the previous section. Indeed, for a timed language C let λC be the same language with all times multiplied by λ (the entropy of this language is $\mathcal{H}(\lambda C) = \mathcal{H}(C) + \log \lambda$). A function ϕ is an (S, C)-encoding with scaling λ if and only if the "λ-scaled" function $\lambda\phi$ is an $(S, \lambda C)$-encoding without scaling. Using this reduction, the results below are corollaries of Prop. 6-7.

Proposition 8. *If for factorial extensible measurable timed language S and sofic timed C an (S, C)-encoding with scaling λ and delay d exists then* (II5) *holds.*

Proposition 9. *If for sofic timed languages S and C (with $\mathcal{H}(S) > -\infty$) the strict version of* (II5) *holds, then there exists an (S, C)-encoding with scaling λ (with some delay d). Moreover it can be realized by a "real-time transducer".*

4.4 A speedup and a slowdown lead to a collapse

For untimed channels, transmission with some rate $\alpha \neq 1$leads to the factor α in information inequalities (II2), (II3). Unfortunately, for timed channels this does not work: any rate $\alpha \neq 1$ leads to a collapse of the previous theory.

Definition 10. *An (S, C)-encoding with rational rate α, delay d and step σ is a function $\phi : S \to C$ such that*

- *its rate is α, i.e. $\forall w \in A^*,\ \lceil \alpha |\phi(w)| \rceil = |w|$;*
- *it is almost injective (with delay d);*
- *no time scaling: its almost inverse ψ is σ-piecewise 1-Lipschitz.*

For $\alpha > 1$ no coding is possible, and for $\alpha < 1$ it always exists. More precisely, the two following propositions hold.

Proposition 10. *For factorial measurable timed languages S and C if $\mathcal{H}(S) > -\infty$ and $\alpha > 1$, then no (S, C)-encoding with rate α exists.*

Proof (Sketch). The proof follows the same lines as that of Prop. 6. Suppose ϕ is such an (S, C)-encoding, and ψ its almost inverse. By Lemma 1, for any natural n we have $S_n \subset \psi_n(C_{\lfloor (n+d)/\alpha \rfloor})$. Each C_n can be covered by C_n^+, a union of $K_{n,\varepsilon}$ cubes of size ε satisfying inequality (1) We have $S_n \subset \psi_n(C^+_{\lfloor (n+d)/\alpha \rfloor})$, and, passing to volumes we get $\mathtt{Vol}(S_n) \leq \mathtt{Vol}(\psi_n(C^+_{\lfloor (n+d)/\alpha \rfloor})) \leq K_{\lfloor (n+d)/\alpha \rfloor, \varepsilon} \varepsilon^n$. Passing to logarithms, dividing by n and taking the limit as $n \to \infty$ we get

$$\mathcal{H}(S) \leq \alpha^{-1} \lim_{n \to \infty} \frac{\log K_{\lfloor (n+d)/\alpha \rfloor, \varepsilon}}{\lfloor (n+d)/\alpha \rfloor} + \log \varepsilon,$$

and applying inequality (1) we obtain

$$\alpha \mathcal{H}(S) \leq \mathcal{H}(C) + \log(1/\varepsilon) + \zeta + \alpha \log \varepsilon = \mathcal{H}(C) + \zeta - (\alpha - 1) \log(1/\varepsilon).$$

Choosing ε small enough makes the inequality wrong. This contradiction concludes the proof. □

Proposition 11. *For sofic timed languages S and C (with $\mathcal{H}(C) > -\infty$) and any $\alpha < 1$ there exists an (S, C)-encoding with rate α (and some delay d). Moreover it can be realized by a kind of timed transducer.*

The construction is non-trivial and uses spare time durations in the channel message to transmit discrete information.

5 Discussion and perspectives

In the previous section, we have established several results on timed channel coding following the standard scheme: a setting of information transmission – information inequality – coding existence theorem – synthesis of an encoder/decoder. We believe that this approach can be applied to various situations of data transmission (and compression). We also consider it as a justification of our previous research on entropy of timed languages [2–4]. In this concluding section, we explain some of our choices and immediate perspectives of this approach.

The time is not preserved. In the central Def. 8 and Prop. 6,7, we consider codings of timed words that preserve the number of events, and not their duration. This choice is compatible to the general idea of dealing with data words (in our case, sequences of letters and real numbers), and less so with the standard timed paradigm. We use again the example of Fig. 2 to illustrate this feature. For $n \in \mathbb{N}$, the timed word $w = [(0.5, e)(0.5, f)]^n$ is encoded to the timed word $w' = [(0.5, a)(1.5, c)]^n$, both have 2 events. However, the duration of w is $(0.5 + 0.5)n = n$, while the duration of w' is $(0.5 + 1.5)n = 2n$.

Other settings to explore. It would be still interesting to explore coding functions preserving durations. On the theoretical side, a more detailed analysis

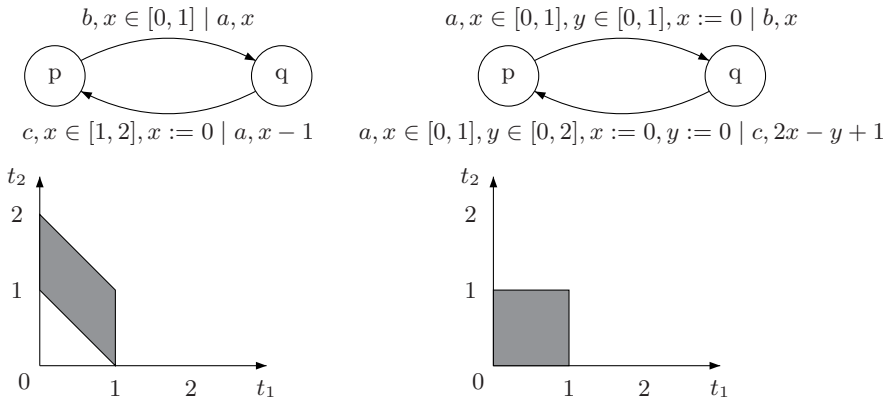

Fig. 4. Top: transducers $\tau_1 : S \to C$ and $\tau_2 : C \to S$. Bottom: languages S_2 and C_2.

for transmission speeds different from 1, as in Sect. 4.4 would probably lead to information inequalities instead of a collapse. Many other settings of transmission of information could be practically relevant: approximated transmission of a timed source on a timed channel; coding of a discrete source on a timed channel; coding using transducers of a fixed precision; coding of other kinds of data languages. On the other hand, more physical models of timed and data channels would be interesting to study, one can think of a discrete channel with a fixed baud rate coupled with an analog channel with a bounded frequency bandwidth. Finally, some special codes, such as sliding-window, error-correcting etc., should be explored for timed and data languages.

What is a timed transducer? In the classical theory of constrained channel coding, several kinds of transducers are used for encoding/decoding, such as those leading to a sliding-block window decoding. In this paper, we have realized our codes by using some very restricted ad hoc timed transducers. They behave like timed (in our case, real-time) automata on the input, and "print" letters and real numbers on the output; we believe that this is the correct approach. However, the right definition of a natural class of timed transducers adequate to coding remains an open question. As a preliminary definition, we suggest timed automata that output, on each transition, a letter and a real number (which is an affine combination of clock values). While reading a timed word, such a transducer would output another timed word. We illustrate this informal definition with the example of mutually inverse transducers τ_1 and τ_2 (encoder and decoder) on Fig. 4. Let us consider a run of the transducer τ_2 on the timed word $(t_1, b)(t_2, c)\ldots$ We start from q with $x = 0, y = 0$, after reading (t_1, b) the value of x and y is t_1, we fire the transition, the output is (t_1, a), we pass in q where we read (t_2, c), the value of x is t_2 and the value of y is $t_1 + t_2$, we fire the transition, the output is $(2t_2 - (t_1 + t_2) + 1, a) = (t_2 - t_1 + 1, a)$, we pass in p etc.

For τ_1, the input language of timed words of length 2 starting from p is $S_2(p) = \{(t_1, b)(t_2, c) \mid t_1 \in [0,1],\ t_1 + t_2 \in [1,2]\}$, while for the second transducer it is $C_2(p) = \{(t_1, a)(t_2, a) \mid t_1 \in [0,1],\ t_2 \in [0,1]\}$. These two languages are

depicted in Fig. 4, they have the same volume. It would be impossible to realize this kind of encoding using "rectangular" transducers as in Sect. 4.

How to improve code synthesis? The encoders in Sect. 4 are not completely satisfactory: they use non-integer guards even when the source and the target language are defined using integer timed automata. We believe that this issue can be avoided using a broader class of transducers as suggested just above.

What about timed symbolic dynamics? The classical theory of constrained-channel coding uses as a convenient terminology, and as a toolset, a branch of the theory of dynamical systems called symbolic dynamics. One of us, in [5], started a formulation of timed languages and automata in terms of symbolic dynamics. Relating it to timed channel coding remains a future work.

Applications. In practice, when transmitting (or storing) information containing discrete events and real-valued data, all the information is first converted to the digital form and next encoded for transmission or storage. Our paradigm combines both steps and, in principle, provides better bounds and codes. However, more research is needed to come up with useful practical codes.

Acknowledgment. We thank the anonymous reviewers for valuable remarks that contributed to substantial improvement of this article.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126, 183–235 (1994)
2. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: Analytic approach. In: FORMATS. pp. 13–27. LNCS 5813 (2009)
3. Asarin, E., Degorre, A.: Volume and entropy of regular timed languages: Discretization approach. In: Concur. pp. 69–83. LNCS 5710 (2009)
4. Asarin, E., Degorre, A.: Two Size Measures for Timed Languages. In: FSTTCS. LIPIcs, vol. 8, pp. 376–387 (2010)
5. Basset, N.: Dynamique Symbolique et Langages Temporisés. Master's thesis, Master Parisien de la Recherche Informatique (2010)
6. Basset, N., Asarin, E.: Thin and thick timed regular languages. In: FORMATS. pp. 113–128. LNCS 6919 (2011)
7. Béal, M.P., Berstel, J., Marcus, B., Perrin, D., Reutenauer, C., Siegel, P.H.: Variable-length codes and finite automata. In: Selected Topics in Information and Coding Theory, chap. 14, pp. 505–584. World Scientific Publishing Company (2010)
8. Berstel, J., Perrin, D., Reutenauer, C.: Codes and Automata, Encyclopedia of Mathematics and its Applications, vol. 129. Cambridge University Press (2009)
9. Blahut, R.E.: Digital Transmission of Information. Addison Wesley (1990)
10. Bojanczyk, M.: Data monoids. In: STACS. LIPIcs, vol. 9, pp. 105–116 (2011)
11. Bouyer, P., Petit, A., Thérien, D.: An algebraic approach to data languages and timed languages. Information and Computation 182(2), 137–162 (2003)
12. Immink, K.: EFMPlus: The coding format of the multimedia compact disc. IEEE Transactions on Consumer Electronics 41(3), 491–497 (1995)
13. Lind, D., Marcus, B.: An Introduction to Symbolic Dynamics and Coding. Cambridge University Press (1995)
14. Marcus, B., Roth, R.M., Siegel, P.H.: Constrained systems and coding for recording channels. In: Handbook of Coding Theory, pp. 1635–1764. North-Holland (1998)